RSAvisual: A Visualization Tool for the RSA Cipher

Jun Tao, Jun Ma
Department of Computer Science
Michigan Technological University
Houghton, MI
{junt,junm}@mtu.edu

Melissa Keranen
Department of Mathematical Sciences
Michigan Technological University
Houghton, MI
msjukuri@mtu.edu

Jean Mayo, Ching-Kuang Shene, Chaoli Wang
Department of Computer Science
Michigan Technological University
Houghton, MI
{jmayo,shene,chaoliw}@mtu.edu

ABSTRACT

This paper describes a visualization tool RSAvisual that helps students learn and instructors teach the RSA cipher. This tool permits the user to visualize the steps of the RSA cipher, do encryption and decryption, learn simple factorization algorithms, and perform some elementary attacks. The demo mode of RSAvisual can be used for classroom presentation and self-study. With the practice mode, the user may go through steps in encryption, decryption, the Extended Euclidean algorithm, two simple factorization algorithms and three elementary attacks. The user may compute the output of each operation and check for correctness. This helps students learn the primitive operations and how they are used in the RSA cipher. The opportunity for self-study provides an instructor with greater flexibility in selecting a lecture pace for the detailed materials. Classroom evaluation was positive and very encouraging.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer science education, information systems education*

General Terms

Algorithms, Security

Keywords

Cryptography, visualization

1. INTRODUCTION

Information security is fundamental to many computer applications and cryptography is the cornerstone on which security solutions are constructed. Landmark textbooks and handbooks have appeared [3, 6, 8, 9] and universities and colleges routinely offer courses in cryptography. The demographics of the students taking these courses can be a challenge for educators. Computer Science students commonly have difficulty dealing with the sophisticated mathematics that cryptosystems are built upon, while Mathematics majors often get lost in the details of the complicated algorithms. Pedagogical research on cryptography is at its very beginning and is scarce. Most studies are about security with minimal focus on cryptography. Visualization did play a role in cryptography. Most of these tools focus on security, rather than the algorithm operation, or have a limited scope. Some of them are Java applets or have simple visual aids while a few offer a visualization framework [2, 5].

This paper describes a tool, RSAvisual, that leverages visualization in order to meet this challenge for the RSA algorithm. RSAvisual is designed to help students understand how the RSA algorithm operates, including encryption, decryption, use of the Extended Euclidean algorithm to calculate the private key, and Fermat and Pollard $p-1$ factorization. RSAvisual is flexible in that it can be used for in-class demonstrations or it can be made available to students for self-study. RSAvisual is one component of a visualization system designed for teaching cryptography to undergraduates [10, 11]. RSAvisual was used in a junior-level introductory cryptography course. The evaluation showed that RSAvisual was effective as a tool for self-study and learning the concepts. Students used it for doing homework, checking details and preparing for the final.

In the following, Section 2 provides the background of our cryptography course, Section 3 presents our visualization tool, Section 4 has a detailed study of our findings from a survey, and Section 5 is our conclusions.

2. COURSE INFORMATION

RSAvisual was used in a cryptography course, MA3203 Introduction to Cryptography, that is offered out of the Department of Mathematical Sciences at Michigan Technological University. It is a junior level course that gives a basic introduction to the field of cryptography. This course covers classical cryptography, the Data Encryption Standard (DES), the Advanced Encryption Standard (AES), the RSA algorithm, discrete logarithms, hash functions, and elliptic curve cryptography. For each cryptosystem, we study how it was designed, why it works, how one may attack the system, and how it has been used in practice.

SIGCSE'14, March 3–8, 2014, Atlanta, GA, USA.
Copyright 2014 ACM 978-1-4503-2605-6/14/03 ...$15.00.
http://dx.doi.org/10.1145/2538862.2538891.

Public key cryptography was a major breakthrough in the field of cryptography. It allows two parties to communicate secretly with each other without first meeting to agree upon a key. Because this advancement in the field was so critical in the history of cryptography, a significant portion of the course is dedicated to the RSA algorithm and discrete logarithms. Although there is a good amount of number theory that is involved in RSA and discrete logarithms, it is all taught within the course. Thus, the course is accessible to all students who are mathematically mature but do not necessarily have a strong background in number theory.

Because the course is offered out of the Department of Mathematical Sciences, there is a heavy focus on the mathematical theory behind each of the algorithms studied. However, it is typically the case that the majority of students taking the course are not Math majors, and many of them come from Computer Science.

Due to some development delay, RSAvisual was introduced near the end of the semester after the RSA algorithm was discussed. It was used as the students were preparing for the final exam. The demonstration mode allowed the instructor to go through examples of the algorithm quickly so that the students were able to see an overview of the algorithm, rather than focusing on details. The instructor also discussed the module that demonstrates the Extended Euclidean algorithm for finding inverses, as well as the module which describes a few simple factoring techniques.

RSAvisual has a practice mode which allows students to test themselves on how well they understand the RSA algorithm. In this mode, students are led through the algorithm while being asked to complete the main components on their own. The next section has all the details of the system.

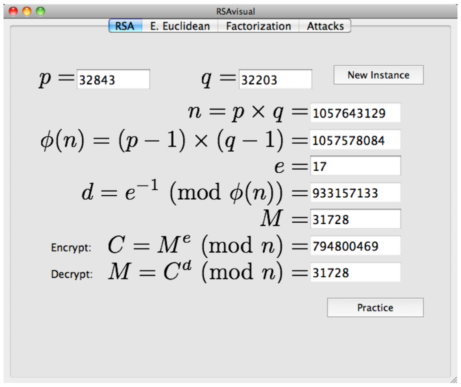
RSAvisual was made available to students to download on their own computer. After the students had access to the software for a week, the instructor distributed a survey to the class. Extra credit was offered to anyone who used the software and completed the survey.

3. SOFTWARE DESCRIPTION

RSAvisual is designed to help students learn the RSA algorithm. It supports Linux, Windows and MacOS. RSAvisual has four components: `RSA`, `E. Euclidean` (Extended Euclidean algorithm), `Factorization` and `Attacks`, each of which corresponds to a page in the system. The `RSA` component has a demo mode and a practice mode. The demo mode shows the details of the computations step by step and is useful in classroom demonstration. The practice mode allows the user to step through the computations, fill in the answers for each step and check for correctness. The two prime numbers p and q are restricted to 5-digit numbers in the demo mode for the user to easily follow the computation steps. Moreover, p and q are restricted to three digits in the practice mode so that the user can perform the computations by hand. RSAvisual always starts from the `RSA` page and the user can switch to other pages freely. The `E. Euclidean` page illustrates the use of the Extended Euclidean algorithm to calculate the inverse of a number. The `Factorization` page demonstrates how to factorize a number with Fermat's algorithm and Pollard's $p-1$ algorithm, respectively. The `Attacks` page has three elementary attacks on the RSA cryptosystem.

3.1 The RSA algorithm

The demo mode of the `RSA` component provides the user with an overall procedure of the RSA algorithm. Given two prime numbers p and q, a public key e, and the plaintext M, it shows how n, $\phi(n)$ and the private key d are computed. Two equations are displayed to show how a sender encrypts the plaintext with public key e and the receiver decrypts that ciphertext with private key d (Figure 1). The user can change the two prime numbers p and q, the public key e and the plaintext M, and the computation will be updated automatically. The user can also click the `New Instance` button to randomly generate a new set of p, q, e and M.



$p =$ 32843 $q =$ 32203

$n = p \times q =$ 1057643129

$\phi(n) = (p-1) \times (q-1) =$ 1057578084

$e =$ 17

$d = e^{-1} \pmod{\phi(n)} =$ 933157133

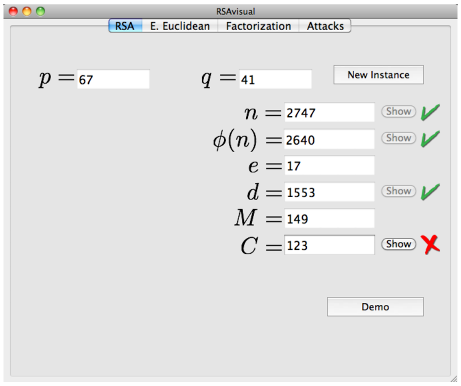
$M =$ 31728

Encrypt: $C = M^e \pmod{n} =$ 794800469

Decrypt: $M = C^d \pmod{n} =$ 31728

Figure 1: Demo mode of the RSA page

In the practice mode, the user can step through the computation (Figure 2); however, all equations are hidden. In each step, a correct result is required to advance to the next step. RSAvisual verifies the input and displays a green tick if the answer is correct. Otherwise, a red cross is shown so that the user can either enter a new value or skip a step by clicking the corresponding `show` button for the system to fill in the correct answer.



$p =$ 67 $q =$ 41

$n =$ 2747 ✓

$\phi(n) =$ 2640 ✓

$e =$ 17

$d =$ 1553 ✓

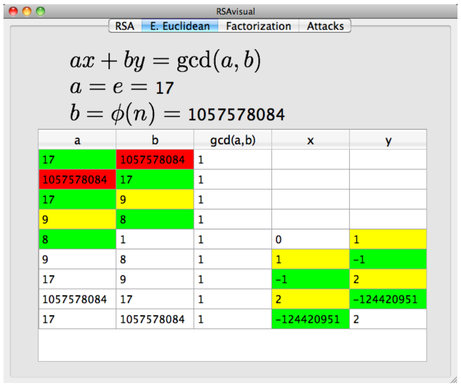
$M =$ 149

$C =$ 123 ✗

Figure 2: Practice mode of the RSA page

3.2 The Extended Euclidean algorithm

The `E. Euclidean` page demonstrates how to compute the inverse of a number using the Extended Euclidean algorithm. Given two integers a and b, it illustrates the computation of x, y and $\texttt{gcd}(a,b)$ in $ax+by = \texttt{gcd}(a,b)$, where $\texttt{gcd}(a,b)$ is the greatest common divisor of a and b. The values of a and b are set to the public key e and $\phi(n)$, respectively, so that x gives the value of private key d. The computation is shown as a table in which each row represents the intermediate results for each step (Figure 3). Two cells of the same color in adjacent rows indicate that the lower one inherits the value from the upper one. The user follows the color of cells to trace the numbers across steps to learn how the values of a and b are exchanged between steps. The values of x and y are not filled bottom-up from where $\texttt{gcd}(a, b)$ is calculated. Instead, they are filled top-down so that it is easier for the user to follow.



$ax + by = \text{gcd}(a, b)$

$a = e =$ 17

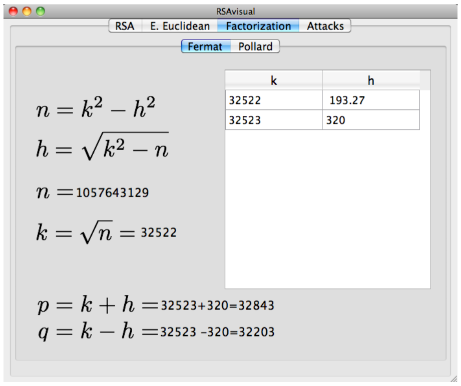
$b = \phi(n) =$ 1057578084

a	b	gcd(a,b)	x	y
17	1057578084	1		
1057578084	17	1		
17	9	1		
9	8	1		
8	1	1	0	1
9	8	1	1	-1
17	9	1	-1	2
1057578084	17	1	2	-124420951
17	1057578084	1	-124420951	2

Figure 3: Computing the Inverse of e Using the Extended Euclidean Algorithm

3.3 Factorization

The **Factorization** component consists of the visualization of two factorization algorithms: Fermat's algorithm and Pollard's $p - 1$ algorithm. They are on two different subpages. Fermat's algorithm starts with $k = \lceil\sqrt{n}\rceil$. At each step, it calculates $h = \sqrt{k^2 - n}$. The values of k and h are recorded and displayed in a table for each step until h is an integer. Finally, the values of p and q are given by $p = k+h$ and $q = k - h$, respectively (Figure 4).



$n = k^2 - h^2$

$h = \sqrt{k^2 - n}$

$n =$ 1057643129

$k = \sqrt{n} =$ 32522

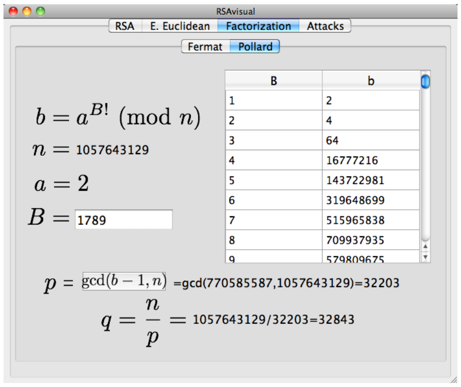
k	h
32522	193.27
32523	320

$p = k + h =$ 32523+320=32843

$q = k - h =$ 32523 -320=32203

Figure 4: Factorizing n Using Fermat's Algorithm

The Pollard algorithm page illustrates how to factorize n by computing $\texttt{gcd}(b - 1, n)$, where $b = a^{B!}$ $(\texttt{mod}\ n)$. When B is large enough, $\texttt{gcd}(b-1, n)$ yields a non-trivial factor of n. The value of B is initialized to be the smallest B with such a property. The value of B can be edited by the user, and RSAvisual will update the value of $\texttt{gcd}(b-1, n)$. In this way, the user will be able to discover that if B is small we have $\texttt{gcd}(b - 1, n) = 1$, and that only if B is large enough $\texttt{gcd}(b - 1, n)$ is a non-trivial factor. The values of B and b are listed in a table, so that the user can verify this property easily (Figure 5). Note that $\texttt{gcd}(b-1, n)$ is a button for the system to show the computation details.



$b = a^{B!}$ (mod n)

$n =$ 1057643129

$a = 2$

$B =$ 1789

B	b
1	2
2	4
3	64
4	16777216
5	143722981
6	319648699
7	515965838
8	709937935
9	579809675

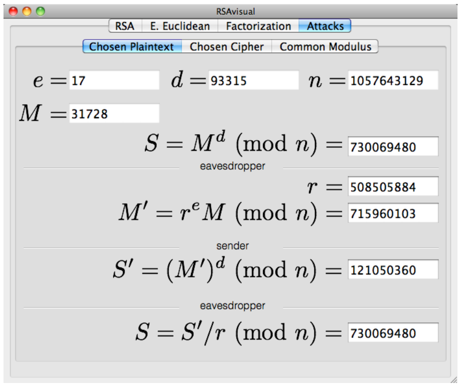
$p =$ gcd$(b-1, n)$ =gcd(770585587,1057643129)=32203

$q = \frac{n}{p} =$ 1057643129/32203=32843

Figure 5: Factorizing n Using Pollard's Algorithm

3.4 Attacks

The **Attacks** component has three elementary attacks on the RSA cryptosystem: chosen plaintext attack, chosen ciphertext attack and common modulus attack [7, 12]. Each of these attacks occupies a sub-page on the **Attacks** page. The same values of e, d, n and M are used and interfaces are similar. For each attack, RSAvisual gives the initial conditions and then displays the attack operations in chronological order. The role of each operation (*i.e.*, sender, receiver and eavesdropper) is specified explicitly (Figure 6, 7 and 8).



$e =$ 17 $d =$ 93315 $n =$ 1057643129

$M =$ 31728

$S = M^d$ (mod n) $=$ 730069480

eavesdropper

$r =$ 508505884

$M' = r^e M$ (mod n) $=$ 715960103

sender

$S' = (M')^d$ (mod n) $=$ 121050360

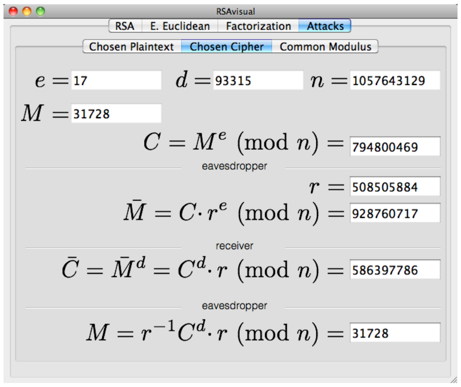
eavesdropper

$S = S'/r$ (mod n) $=$ 730069480

Figure 6: Chosen Plaintext Attack to Forge the Signature of the Sender

4. EVALUATION

The RSAvisual survey consists of two components, a set of 10 questions (Table 1) and 11 write-in comments. The three elementary attacks on RSA were not evaluated because this component was added after the survey was done. Choices available are 5:strongly agree, 4:agree, 3:neutral, 2:disagree, and 1:strongly disagree. Because we intend to study the impact of RSAvisual on multiple disciplines, students were asked to fill in their disciplines. We collected 27 survey forms. The distribution of majors was as follows: 3 in computer network and system administration (CNSA), 10 in electrical and computer engineering (EE/CpE), 9 in computer science (CS and Computer Systems), 3 in mathematics (Math), 1 in Service System Engineering, and 1 undeclared.



$e =$ 17 $d =$ 93315 $n =$ 1057643129

$M =$ 31728

$C = M^e \pmod{n} =$ 794800469

eavesdropper

$r =$ 508505884

$\bar{M} = C \cdot r^e \pmod{n} =$ 928760717

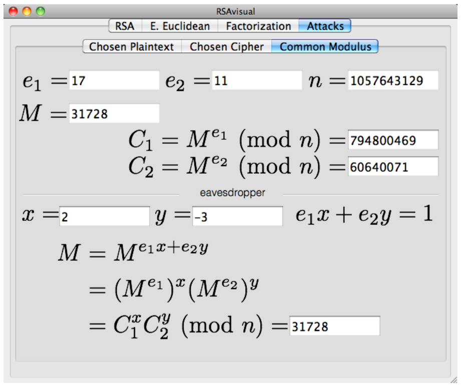
receiver

$\bar{C} = \bar{M}^d = C^d \cdot r \pmod{n} =$ 586397786

eavesdropper

$M = r^{-1} C^d \cdot r \pmod{n} =$ 31728

Figure 7: Chosen Ciphertext Attack to Recover the Plaintext



$e_1 =$ 17 $e_2 =$ 11 $n =$ 1057643129

$M =$ 31728

$C_1 = M^{e_1} \pmod{n} =$ 794800469

$C_2 = M^{e_2} \pmod{n} =$ 60640071

eavesdropper

$x =$ 2 $y =$ -3 $e_1 x + e_2 y = 1$

$M = M^{e_1 x + e_2 y}$

$= (M^{e_1})^x (M^{e_2})^y$

$= C_1^x C_2^y \pmod{n} =$ 31728

Figure 8: Common Modulus Attack to Recover the Plaintext

4.1 General Discussion

Table 2 shows the mean and standard deviation of each question. In general, reactions to RSAvisual were positive. The highest score of 4.1 with a standard deviation 0.7 was given to Q1, indicating that students agreed highly that RSAvisual helped them understand the RSA cipher. Q2, Q5, Q6 and Q10 received scores no less than 3.9, suggesting that RSAvisual enhanced self-study and the course. The two factorization related questions, Q3 and Q4, were rated at 3.4, the lowest score of all questions. This may be due to the use of variable names (Section 4.3). Thus, while Q3 and Q4 received lower rating, the general trend of student reactions was still on the positive side.

The ratings of each question are loosely positive related with the highest correlation being 0.65 for the (Q1,Q10) pair and 0.63 for the (Q1,Q2) pair although both are not very high (Table 3). This suggests that those who believed the demo mode was helpful tended to rate the demo mode higher and considered that RSAvisual enhanced the course. Students who understood the RSA algorithm more after using the tool (Q8) also believed the demo mode helped them understand the RSA algorithm (Q1), the Extended Euclidean algorithm (Q2) and the Fermat Factorization algorithm (Q3), although the correlation coefficients are between 0.52 and 0.57. Pairs (Q2,Q6), (Q3,Q6), (Q4,Q6), (Q4,Q8) and (Q4,Q9) have very low but negative correlation coefficients. The Q6 pairs are understandable because Q6 is about encryption and decryption that are irrelevant to the Extended Euclidean and factorization algorithms (Q2, Q3 and Q4). The (Q4,Q8) and (Q4,Q9) pairs share the same reasoning. However, the (Q4,Q7) pair with a correlation of −0.31 is somewhat puzzling, suggesting that even though students believed RSAvisual helped them understand the RSA algorithm better, they tended to rate the Pollard's $p-1$ factorization algorithm visualization in a slightly negative way.

Table 1: Survey Questions

No.	Question
Q1	RSAvisual 's demo mode helped me understand
Q2	RSAvisual 's demo mode helped me better understand the Extended Euclidean algorithm
Q3	RSAvisual 's demo mode helped me better understand the Fermat Factorization algorithm
Q4	RSAvisual 's demo mode helped me better understand the Pollard $p-1$ Factorization algorithm
Q5	RSAvisual 's demo mode was helpful for my self-study
Q6	RSAvisual 's practice mode helped me remember how to encrypt and decrypt with the RSA algorithm
Q7	RSAvisual 's practice mode helped me understand how to find inverses using the Extended Euclidean algorithm
Q8	I understand the RSA algorithm more after using RSAvisual
Q9	By using RSAvisual I was able to identify the parts of the RSA algorithm that I did not understand
Q10	RSAvisual enhanced the course.

Table 2: Mean μ and Standard Deviation σ

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
μ	4.1	3.9	3.4	3.4	4.0	3.9	3.7	3.8	3.6	3.9
σ	0.7	0.9	0.9	0.8	0.7	0.8	0.8	0.7	0.7	0.7

Table 3: Correlation Matrix of Questions

	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Q1	0.63	0.49	0.35	0.44	0.30	0.07	0.57	0.32	0.65
Q2		0.57	0.28	0.23	-0.07	0.44	0.52	0.32	0.40
Q3			0.56	0.28	-0.07	0.12	0.54	0.10	0.42
Q4				0.19	-0.13	-0.31	-0.06	-0.02	0.40
Q5					0.47	0.06	0.45	0.35	0.26
Q6						0.08	0.20	0.42	0.06
Q7							0.46	0.29	0.13
Q8								0.36	0.37
Q9									0.44

4.2 Statistical Analysis

The differences among student groups were studied using ANOVA for each question. Since the questions may correlate with each other, we also applied MANOVA (Multivariate ANOVA) to investigate the overall differences. For MANOVA, Wilk's Lambda test was used and all questions were considered at the same time. The students were grouped by their majors: CS, EE, CpE, Csys (Computer Systems), CNSA, Math and Other. The group Other contained a student who did not provide his discipline information and a student who majored in Service System Engineering. Although the MANOVA result suggested significant differences among student groups with p-value of 0.032, we did not find differences at significance level of 0.05 for any question using ANOVA. The smallest p-value in ANOVA results was 0.077 for Q7. The p-values for the other questions were all larger than 0.231. This suggests that the rating of students from different disciplines did not vary significantly.

To better understand the possible differences among students, we applied cluster analysis to group the students. The Ward's method with the Mahalanobis distance was used for a hierarchical agglomerative clustering and two groups were formed as shown in Figure 9. MANOVA showed that the two groups of students rated very differently (p-value = 0.00006). ANOVA for each individual question also found significant differences for Q4 and Q5 with p-values of 0.014 and 0.005, respectively. The mean values of rating from these two groups revealed that the students in Group 1 usually offered higher ratings than those in Group 2, except for Q6 (3.875 against 4) and for Q10 (3.875 against 3.909), which were still close. Students in Group 1 rated the first five questions 0.4 higher than Group 2. However, no clear and significant evidence, such as disciplines, can be found to explain these differences. It was more likely that the higher rates of Group 1 were just a personal preference rather than discipline-related. This was also consistent with our findings from the results where the students were grouped by majors.

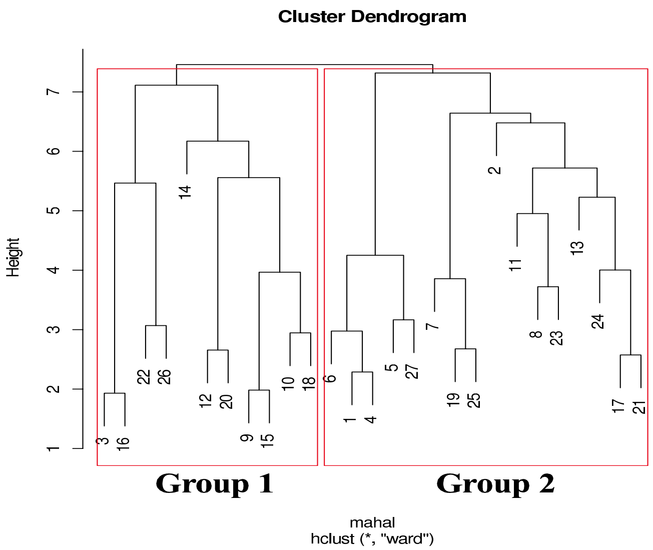


Figure 9: Cluster Analysis Result

4.3 Student Comments

The set of 11 write-in questions is designed to allow students to make suggestions which can be used for future development. We focus on the following issues: the restriction of using 5-digit numbers, whether the Extended Euclidean algorithm needs improvement, whether the display/use of Fermat Factorization is good enough, whether the display/use of Pollard Factorization is good enough, the evaluation of the demo and practice modes, the evaluation of the practice mode, frequency of using RSAvisual for self-study, and software installation issues.

Most of the student comments did not consider the use of 5-digit numbers a limitation. Typical comments were "*I think only 5 digits was helpful because the smaller numbers were easier to try by hand*", "*For the purpose of example problems, this was not a significant limitation*" and "*No, the concept is what is important here*". One student said "*It was okay, but didn't let me try all the examples I worked*". On the other hand, another student indicated "*I find it easier to use 2 digits when playing with the visualization software*". Therefore, we believe that the use of 5-digit numbers is a very reasonable restriction.

Students were positive about the Extended Euclidean module, the Fermat Factorization module and the Pollard module. Typical comments on the Extended Euclidean module were "*This module provided me with a simplified way to perform the Extended Euclidean algorithm*", "*I thought it was easy enough as it seemed to be useful for understanding/grasping the concept*" and "*Helpful for reviewing the Extended Euclidean algorithm. It showed a faster way to find inverses*". Due to the many ways of performing a typical Euclidean algorithm (*i.e.*, recursive, tabular and formula substitution), some students were initially confused about the tabular presentation. However, students also noted that after a classroom presentation, they were able to understand the content of the table and felt it was easy to follow. The Fermat and Pollard modules followed the same trend. Most students felt that both modules were "*straightforward*", "*easy enough to use and understand*" and "*illustrative of concept*". A handful of students were not used to the tabular presentation even though the simple formulas were shown next to the table. Some students complained about the use of variable names being different from those in the textbook. We do not think this is a significant issue because different textbooks may use different variable names.

The practice mode was well received with comments like "*The practice mode is great! I like that it is step by step and I can try any instance I want*", "*The program is of most use as a way to check your work on RSA problems*", "*Simple and straight-forward*", "*When you input an incorrect answer the program does not automatically display the correct answer and lets you re-enter another answer without starting over. This helps me to really work out the solutions multiple times if needed*" and "*Good, easy tool for determining ability level of using the algorithm*". Students also offered suggestions to improve the practice mode. In addition to font size and spacing adjustments, a few students wished to see hints and/or explanations of each step so that their understanding can be reinforced.

The demo mode was not universally welcome because some students considered that the demo mode would replace blackboard work. Along this line of thinking, they would prefer blackboard work because the pace was slower allowing the instructor to show the detailed work step by step. This may be due to the fact that the RSA algorithm was presented in class as a lecture, but RSAvisual was demonstrated weeks later during the final exam review. Therefore, when the students saw the demo mode in action, it was presented quickly because it had already been covered in detail. Students may have had a different reaction if RSAvisual had been introduced along with the backboard presentation of the algorithm. On the other hand, the majority of students indicated that the demo mode was helpful and correctly pointed out that "*The two parts together were definitely necessary*", "*Wouldn't want one without the other*" and "*I would not say that the program should replace the blackboard but it is a good supplement*". Typical comments were "*The most effective part was that you could clearly see all of the steps and calculations at the same time*" and "*I really liked how you could enter your own values and that it went through the steps*". Major improvement suggestions include adding hints and explanations when needed and providing an option so that the computation would be shown step by step under the control of the user. We have taken all of these suggestions into consideration and are working on an update.

Students indicated that "*I really appreciate the simplicity of the layout*" and "*Layout was simple*". The most needed improvements include font size and space issues, a mechanism to activate hints, formulas and descriptions, variable speed and step-by-step execution in the demo mode and some way to tell the user what went wrong about the input and intermediate values in the practice mode.

Students only had a week to play with RSAvisual before taking this survey and the frequency of using this tool is not very high. Most of them used the tool a few times, and a few played with the tool "often". They used RSAvisual for solving problems, checking for details, and preparing for the final. Students reported some installation issues such as missing shared libraries, mostly from MacOS users as they perhaps did not install the needed libraries properly. Windows 7 and Vista users did not report major installation issues.

In summary, we believe RSAvisual has helped students learn and the instructor teach the RSA cipher effectively. With the comments and suggestions, we should be able to improve RSAvisual significantly in the near future.

5. CONCLUSIONS

This paper presented a visualization tool RSAvisual for teaching and learning the RSA algorithm, the Extended Euclidean algorithm for finding inverses, Fermat's and Pollard's factorization algorithms, and some elementary attacks. With RSAvisual, instructors are able to present all the details and inner working of these algorithms. It also helps students see the "flow" of each algorithm, learn the concepts and practice the computation steps using the practice mode. Evaluation results showed that RSAvisual was effective in classroom presentation and for student self-study. While students in this class were from more than five disciplines, our analysis did not find clear and significant evidence of discipline-specific rating differences. Thus, although a cluster analysis grouped the student ratings into two clusters, we concluded that the rating differences may be due to a pure personal preference.

RSAvisual only uses five digits in the demo mode and three digits in the practice mode, and implements the "textbook" version of the RSA algorithm. While nearly all students indicated these restrictions worked fine, we are looking at ways to extend RSAvisual and alleviate these limits. In addition to font size and spacing issues, we plan to **(1)** increase the number of digits, which is user selectable, so that the system can still be easy to use, **(2)** include a padding module, **(3)** consider better or more visualization styles for the Extended Euclidean and factorization algorithms, and **(4)** add a few more elementary attacks on RSA.

We are also investigating some recent advances in prime number related research results and hope to determine if they are worthwhile to be mentioned for our course. It has been known for decades that determining if a positive integer is a prime (*i.e.*, **PRIMES**) is in $\mathcal{NP} \cap \text{co-}\mathcal{NP}$. However, Manindra Agrawal, Neeraj Kayal and Nitin Saxena (AKS) proved in 2004 that **PRIMES** is actually in $\mathcal{P}$ [1, 4]. A more recent and surprising result was proved by Yitang Zhang in 2013, indicating the gap between two consecutive prime numbers is less than 7×10^7 [13]. Whether these results can contribute to or be used to the development of fast factorization algorithms is under investigation.

RSAvisual is a part of larger development of cryptography visualization tools supported by the National Science Foundation. In addition to RSAvisual, DESvisual for the DES cipher and ECvisual for the elliptic curve based ciphers are available online and new tools SHAvisual and AESvisual for SHA and AES will become available soon. Tools, evaluation forms, and installation and user guides for Linux, MacOS and Windows can be found at the following URL:

`www.cs.mtu.edu/~shene/NSF-4`.

6. REFERENCES

[1] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annual of Mathematics*, 160:781–793, 2004.

[2] G. Cattaneo, A. D. Santis, and U. F. Petrillo. Visualization of Cryptographic Protocols with GRACE. *Journal of Visual Languages and Computing*, 19:258–290, April 2008.

[3] D. E. R. Denning. *Cryptography and Data Security.* Addison-Wesley, 1982.

[4] M. Dietzfelbinger. *Primality Testing in Polynomial Time: from Randomized Algorithms to "PRIMES is in P".* Springer-Verlag, 2004.

[5] D. Ebeling and R. Santos. Public Key Infrastructure Visualization. *Journal of Computing in Small Colleges*, 23:247–254, October 2007.

[6] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.

[7] R. A. Mollin. *RSA and Public-Key Cryptography.* Chapman & Hall/CRC, 2003.

[8] A. Salomaa. *Public-Key Cryptography.* Springer-Verlag, 1992.

[9] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* Wiley, second edition, 1995.

[10] J. Tao, J. Ma, M. Keranen, J. Mayo, and C.-K. Shene. DESvisual: A Visualization Tool for the DES Cipher. *Journal of Computing Science in Colleges*, 27:81–89, October 2011.

[11] J. Tao, J. Ma, M. Keranen, J. Mayo, and C.-K. Shene. ECvisual: A Visualization Tool for the Finite Field Elliptic Curve Cipher. In *Proceedings of the ACM Technical Symposium on Computer Science Education*, pages 571–576, 2012.

[12] S. Y. Yan. *Cryptanalytic Attacks on RSA.* Springer-Verlag, 2008.

[13] Yitang Zhang. Bounded Gaps between Primes. *Annual of Mathematics.* (to appear).

Acknowledgment

The authors are supported by the National Science Foundation under grants DUE-1140512, DUE-1245310 and IIS-1319363.